**SOFTWARE CONSTRUCTION (PRACTICALS) – SPRING 2013**

**MINI PROJECT 01 – PARSER, SEMANTIC ANALYSER AND CODE GENERATOR SIXTEEN ++**

|  |  |  |  |
| --- | --- | --- | --- |
| **COURSE:** BESE 16 A & B | | **MAXIMUM MARKS:** 30 | |
| **DEADLINE: 20th May 2013** | | **INSTRUCTORS:** Dr. Zaki Murtaza  Lec Ayesha Nasser  Engr. Umar Mahmud | |
|  | **Instructions**   * Only **TWO** students per syndicate are allowed. You have to keep the same group as an Mini- Project 01 * This is a project assignment and is based on the lab experiments carried out by you. * Plagiarism is strictly forbidden. * You have to create the project for the given language specification. * You have to show the working project for evaluation before deadline. * The evaluation is based on project demonstration, working of the project for different test patterns and associated viva. | |  |
|  | **Sixteen++:** Department of Computer Software Engineering is a pioneer department in software engineering in Pakistan with over 15 years of continued service in training software engineers. The department wishes to develop a programming language and its associated compiler. The tool will serve as a training agent for future courses. The language is named as Sixteen++ that follows a C++ style programming. The language is in its initial phases of development and is a structured language. You as part of BESE 16 are required to create a lexical analyser for Sixteen++ following the given specifications. | |  |
| 1. | **Objectives:**   1. Parser 2. Semantic Analyser 3. Code Generator. | |  |
| 2. | **Programming Language:** C++/C#/Java/VB/Python | |  |
| 3. | **Software Required:** NIL | |  |
| 5. | **Data Types:** There are only four data types. The data types are as under:-   1. **number**  to represent an integer number 2. **rational** to represent a rational number (decimal point number) 3. **truth** to represent true or false 4. **letter** to represent any character | |  |
| 6. | **Variable Declaration Statements:** Variable declaration is given as  ***var* space variable-name space colon space datatype? space semicolon**  For example,   * 1. var int1 : number? ;   2. var float1 : rational? ; | |  |
| 7. | **Operators:**   1. **|+** for addition 2. **|-** for subtraction 3. **|\*** for multiplication 4. **|/** for division 5. **|%** for remainder 6. **<>** for concatenation 7. **!**  for NOT logic 8. **|&** for AND logic 9. **|#** for OR logic 10. **[ ]** for function body 11. **( )** for precedence and grouping operations 12. Precedence is same as that of C++ 13. |< for less than 14. |<= for less than or equal to 15. |> for greater than 16. |>= for greater than or equal to 17. |== for is equal to | |  |
| 8. | **Functions and Keywords:** Functions and keywords for loop and conditional statements are provided at the moment: -   1. ***acquire*** *f*or acquiring single Byte input 2. ***display*** *f*or displaying single Byte output 3. **if** for if statments 4. **elif** for else-if statements 5. **el** for else 6. **fi** for end of if 7. **repeat i : 10, 1** is a for statement with i as a variable, 10 is the upper limit and 1 as the lower limit | |  |
| 9. | **Sample Program:**  A sample program is given as under:-  . [  . var int1 : whole ;  . acquire int1 ;  . display int1 ;  . ]  1  2  3  4  5   1. Each line starts with a . 2. Line 1 should always contain [ 3. ‘acquire’ statement takes a single Byte of input 4. ‘display’ statement displays a single Byte of output 5. Last line should always contain ]   . [  . var my\_int: number  . var my\_float: rational  . repeat I:10,1  . my\_int = acquire(number)  . my\_float = acquire(rational)  . if ( my\_int < my\_float)  . display(number, my\_int)  . elif (my\_int > my\_float)  . display(rational, my\_float)  . fi  . ] | |  |
| 10. | You are required to identify whether the given code is correct or not, create lexeme table, symbol table and parse tree also you have generate the target code. Try to have the target code in assembly language that can be executed. | |  |
| 11. | The test involves a viva and grading of the project based on different codes. Start doing what you can with current knowledge and use the practices from the future labs to do or modify your work on this project. | |  |